

FILEID**RSE

D 2

RRRRRRRR		SSSSSSSS	EEEEEEEEE
RRRRRRRR		SSSSSSSS	EEEEEEEEE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RRRRRRRR		SSSSSS	EEEEEEEEE
RRRRRRRR		SSSSSS	EEEEEEEEE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SSSSSSSS	EEEEEEEEE
RR	RR	SSSSSSSS	EEEEEEEEE

....

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

RSE
V04

(1)	40	HISTORY	: DETAILED
(1)	78	DECLARATIONS	
(1)	137	SCH\$RSE - REPORT SYSTEM EVENT	
(1)	270	SCH\$UNWAIT - DECREMENT COUNT IN WAIT QUEUE	
(1)	323	SITUATIONAL PRIORITY INCREMENT TABLE	
(1)	342	SCH\$CHSE - CHANGE STATE TO EXECUTABLE	
(1)	439	SWPO - SWAP OUT SIMPLE NON-EXECUTABLE	
(1)	457	SCH\$QEND - QUANTUM END ROUTINE	
(1)	612	SENDAST - Send AST to process	
(1)	659	SCH\$WAKE - WAKE PROCESS INTERNAL	
(1)	697	SCH\$SWPWAKE - WAKE SWAPPER PROCESS	

0000 1 .TITLE RSE - REPORT SYSTEM EVENT
0000 2 .IDENT 'V04-000'
0000 3 .
0000 4 .
0000 5 .*****
0000 6 .
0000 7 .** COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 .** DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 .** ALL RIGHTS RESERVED.
0000 10 .
0000 11 .** THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 .** ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 .** INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 .** COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 .** OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 .** TRANSFERRED.
0000 17 .
0000 18 .** THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 .** AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 .** CORPORATION.
0000 21 .
0000 22 .** DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 .** SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 .
0000 25 .
0000 26 .*****
0000 27 .
0000 28 .++
0000 29 .FACILITY: EXECUTIVE, SCHEDULER
0000 30 .
0000 31 .ABSTRACT:
0000 32 . THIS MODULE CONTAINS THE SYSTEM EVENT REPORTING ROUTINES AND
0000 33 . THEIR SUPPORTING SUBROUTINES.
0000 34 .
0000 35 .ENVIRONMENT:
0000 36 . MODE = KERNEL
0000 37 .--
0000 38 .

0000 40 .SBTTL HISTORY ; DETAILED
0000 41 :
0000 42 : AUTHOR: R. HUSTVEDT CREATION DATE: 6-SEP-76
0000 43 :
0000 44 : V03-008 SSA0015 Stan Amway 8-Mar-1984
0000 45 : Allow expansion of working set if PFRATH exceeded and
0000 46 : number of active pages (PPG + GPG) exceeds 75% of WSSIZE.
0000 47 : Previously, expansion was done only if the number of active
0000 48 : pages was equal to WSSIZE.
0000 49 : (Acknowledgements go to Wayne Cardoza and Larry Kenah,
0000 50 : who both collaborated on this change.)
0000 51 :
0000 52 : V03-007 WMC0002 Wayne Cardoza 28-Feb-1984
0000 53 : Fix checks for waking swapper.
0000 54 :
0000 55 : V03-006 LY00B4 Larry Yetto 10-FEB-1984 10:29
0000 56 : Fix truncation errors
0000 57 :
0000 58 : V03-005 TMK0002 Todd M. Katz 27-Dec-1983
0000 59 : Fix broken branches.
0000 60 :
0000 61 : V03-004 SSA0003 Stan Amway 5-Dec-1983
0000 62 : Added support for outswap scheduling changes.
0000 63 : Record event time for process unwait and quantum end
0000 64 : event.
0000 65 :
0000 66 : V03-003 TMK0001 Todd M. Katz 13-Nov-1983
0000 67 : Fix broken branches.
0000 68 :
0000 69 : V03-002 TCM0001 Trudy C. Matthews 4-Apr-1983
0000 70 : Change references to working set fields in PHD so that
0000 71 : they are used as unsigned words.
0000 72 :
0000 73 : V03-001 WMC0001 Wayne Cardoza 11-Mar-1983
0000 74 : Bad comparison against BORROWLIM.
0000 75 :
0000 76 :

```
0000 78 .SBTTL DECLARATIONS
0000 79
0000 80 :
0000 81 : INCLUDE FILES:
0000 82 :
0000 83     $ACBDEF          : DEFINE AST CONTROL BLOCK
0000 84     $DYNDEF          : DEFINE STRUCTURE TYPE CODES
0000 85     $CEBDEF          : DEFINE COMMON EVENT BLOCK
0000 86     $IPLDEF           : IPL DEFINITIONS
0000 87     $PCBDEF           : PCB DEFINITIONS
0000 88     $PHDDEF           : PROCESS HEADER DEFINITIONS
0000 89     $PRDEF             : PROCESSOR REGISTER DEFS
0000 90     $PRIDEF            : PRIORITY INCREMENT CLASSES
0000 91     $SSDEF              : DEFINE STATUS CODES
0000 92     $STATEDEF          : STATE DEFINITIONS
0000 93     $WQHDEF             : WAIT QUEUE HEADER DEFINITIONS
0000 94 :
0000 95 : MACROS:
0000 96 :
0000 97     .MACRO EVENT,EVTN,STATLIST,EACTION,CONT=0
0000 98     .IF NB,EVTN
0000 99 EVT$_`EVTN==EVTCTR
0000 100     .ENDC
0000 101 EVTCTR=EVTCTR+1
0000 102     .WORD EACTION-STACT
0000 103 RSE...=.
0000 104     .PSECT AES2,BYTE
0000 105 STMSK=CONT
0000 106     .IRP ST,<STATLIST>
0000 107 STMSK=STMSK+<1@SCH$C_ST>
0000 108     .ENDR
0000 109     .LONG STMSK
0000 110     .PSECT AES1,BYTE
0000 111 .=RSE...
0000 112     .ENDM EVENT
0000 113 :
0000 114 :
0000 115 : GENERATE MASK FOR WAIT STATES
0000 116 :
0000 117 : GMASK STATENAME
0000 118 :
0000 119 :
0000 120     .MACRO GMASK,STATE
0000 121 ST=SCH$C_STATE
0000 122 WAITST=WAITST+<1@ST>
0000 123     .ENDM GMASK
0000 124 :
0000 125 : EQUATED SYMBOLS:
0000 126 :
0000 127 EVTCTR=0          : INITIALIZE EVENT COUNTER TO 0
0000 128 WAITST=0           : INITIALIZE WAIT STATE MASK
0000 129 ASTEXIT=0          : AST EXIT CHANGE MODE CODE
0000 130 :
0000 131 : OWN STORAGE:
0000 132 :
0000 133     .PSECT AES2,BYTE          : STATE EVENT MASK PSECT
0000 134 STET= .               : BASE OF STATE EVENT TABLE
```

RSE
V04-000

- REPORT SYSTEM EVENT
DECLARATIONS

I 2

16-SEP-1984 01:06:34 VAX/VMS Macro V04-00
5-SEP-1984 03:47:04 [SYS.SRC]RSE.MAR;1

Page 4
(1)

00000000 135

.PSECT AES1,BYTE

RSE
V04

0000 137 .SBTTL SCH\$RSE - REPORT SYSTEM EVENT
 0000 138
 0000 139 :++
 0000 140 : FUNCTIONAL DESCRIPTION:
 0000 141 : SCH\$RSE RECEIVES SYSTEM EVENT REPORTS FROM VARIOUS SOURCES
 0000 142 : AND PERFORMS THE APPROPRIATE ACTION FOR THE SPECIFIED PROCESS.
 0000 143 : EVENT REPORTING MUST BE PERFORMED WITH IPL=IPL\$ SYNCH.
 0000 144 : AS A SIDE EFFECT OF AN EVENT REPORT, THE RESCHEDULING INTERRUPT
 0000 145 : MAY BE TRIGGERED IF APPROPRIATE.
 0000 146
 0000 147 : CALLING SEQUENCE:
 0000 148 : BSB/JSB SCH\$RSE
 0000 149 : .BYTE EVTS_EVENTNAME
 0000 150
 0000 151 : THIS CALLING SEQUENCE IS GENERATED BY THE RPTEVT SYSTEM MACRO
 0000 152
 0000 153 : REPEVT EVENTNAME
 0000 154
 0000 155 : INPUT PARAMETERS:
 0000 156 : R2 - SITUATIONAL PRIORITY INCREMENT CLASS NUMBER
 0000 157 : R4 - PCB ADDRESS OF PROCESS FOR WHICH EVENT IS REPORTED
 0000 158
 0000 159 : EVENT NUMBER CONTAINED IN BYTE LOCATED BY ADDRESS AT TOP
 0000 160 : OF STACK. @SP
 0000 161
 0000 162 : IMPLICIT INPUTS:
 0000 163 : SCHEDULER DATA BASE
 0000 164
 0000 165 : OUTPUT PARAMETERS:
 0000 166 : NONE
 0000 167
 0000 168 : IMPLICIT OUTPUTS:
 0000 169 : NONE
 0000 170
 0000 171 : COMPLETION CODES:
 0000 172 : NONE
 0000 173
 0000 174 : SIDE EFFECTS:
 0000 175 : A RESECHEDULING INTERRUPT MAY BE REQUESTED IF THE SPECIFIED
 0000 176 : PROCESS IS HIGHER IN PRIORITY THAN THE CURRENT PROCESS.
 0000 177
 0000 178 :--
 0000 179

				180 SCH\$RSE::	
				181 MOVZBL @SP,R3	: REPORT SYSTEM EVENT
				182 INCL (SP)	: GET EVENT NUMBER
				183 MOVZWL PCB\$W STATE(R4),R1	: UPDATE RETURN ADDRESS
50	51 2C A4	3C	0006	184 10\$: MOVL W\$TET[R3],R0	: GET CURRENT STATE NUMBER
	0000'CF43	DO	000A	185 BBS R1,R0,ACTION	: GET STATE MASK FOR EVENT
	06 50	51	E0	186 INCL R3	: DO ACTION IF STATE BIT SET
				187 BLBS R0,10\$: CHECK NEXT ACTION
	F1 50	E8	0016	188 RSB	: IF CONTINUATION
				189	: OTHERWISE IGNORE EVENT
	0A' 00 53	CF	001A	190 ACTION: CASEL R3,#0,S^#MAXEVT	: SWITCH ON EVENT NUMBER(UPDATED)
				191 STACT: EVENT AST,<-	: BASE OF ACTION TABLE
				192 CEF,-	: AST EVENT
				193	: COMMON EVENT FLAG WAIT

001E	194	COLPG,-	: COLLIDED PAGE WAIT
001E	195	FPG,-	: FREE PAGE WAIT
001E	196	HIB,-	: RESIDENT HIBERNATE
001E	197	HIBO,-	: NON-RESIDENT HIBERNATE
001E	198	LEF,-	: LOCAL EVENT FLAG WAIT
001E	199	LEFO,-	: LOCAL EVENT FLAG WAIT (NON-RES)
001E	200	MWAIT,-	: MUTEX WAIT
001E	201	PFW,-	: PAGE FAULT WAIT
001E	202	>,EVENTE	: AST EXECUTABLE STATE CHANGE
00000000	0020	EVTS_COLPGA==EVTS_AST	: USE SAME EVENT FOR COLLIDED PAGE AVAIL
0020	204		
0020	205		
0020	206	EVENT EVENT,<LEF>,LEFEVT,CONT=1	: EVENT FLAG SETTING
0022	207	,<CEF>,CEFEVT,CONT=1	: COMMON EVENT FLAG SET
0024	208	,<LEFO>,EVENTE	: NON-RESIDENT LOCAL EVENT
0026	209		
0026	210	FGPA,<-	: FREE PAGE AVAILABLE
0026	211	FPG,-	: FREE PAGE WAIT STATE
0026	212	>,EVENTF	: EXECUTABLE STATE CHANGE
0028	213	WAKE,<-	: WAKE EVENT
0028	214	HIB,-	: RESIDENT HIBERNATE
0028	215	HIBO,-	: NON-RESIDENT HIBERNATE
0028	216	>,EVENTE	: EXECUTABLE
002A	217		
002A	218	EVENT RESUME,<-	: RESUME EVENT
002A	219	SUSP,-	: RESIDENT SUSPENDED
002A	220	SUSPO,-	: NON-RESIDENT SUSPENDED
002A	221	>,EVENTE	: EXECUTABLE
002C	223		
002C	224		
002C	225	EVENT PFCOM,<-	: PAGE FAULT COMPLETE EVENT
002C	226	PFW,-	: PAGE FAULT WAIT
002C	227	>,EVENTE	: EXECUTABLE
002E	228		
002E	229	EVENT SETPRI,<-	: SET PRIORITY EVENT
002E	230	COM,-	: RESIDENT COMPUTE
002E	231	COMO,-	: NON-RESIDENT COMPUTE
002E	232	>,EVENTF	: EXECUTABLE
0030	233		
0030	234	SWPOUT,<-	: SWAP OUT EVENT
0030	235	HIB,-	: RESIDENT HIBERNATE
0030	236	LEF,-	: RESIDENT LOCAL EVENT FLAG WAIT
0030	237	SUSP,-	: RESIDENT SUSPENDED
0030	238	>,SWPO,CONT=1	: SIMPLE SWAP OUT
0032	239	EVENT <-,	: SWAP OUT EVENT CONINUATION
0032	240	COM,-	: RESIDENT COMPUTE
0032	241	>,SWPOE	: EXECUTABLE OUTSWAP
0034	242		
0034	243	:	
0034	244	IF CASE FALLS THROUGH, THEN BUGCHECK WITH ILLEGAL EVENT	
0034	245	NUMBER.	
0034	246	:	
0000000A	0034	MAXEVT=EVTCTR-1	: MAXIMUM EVENT NUMBER
0034	248	BUG_CHECK ILLEVNUM,FATAL	: ILLEGAL EVENT NUMBER
0038	249		

0038 251 :
0038 252 :
0038 253 :
0038 254 :
OE 24 A4 E9 0038 255 :
53 6C A4 D0 003C 256 :
00C0 C3 04 C0 0040 257 :
0088 C3 01 3C 0045 258 :
004A 259 :
004A 260 :
004A 261 :
004A 262 :
004A 263 :
004A 264 :
88'AF 9F 004A 265 :
004D 266 :
004D 267 :
004D 268 :
COMMON EVENT FLAG SET
CEFEVT: BLBC PCB\$L_STS(R4),EVENTE ; BR IF NOT RESIDENT
LEFEVT: MOVL PCB\$L_PHD(R4),R3 ; LOCAL EVENT FLAG SET FOR RESIDENT PROCESS
ADDL #4,PHD\$L_PC(R3) ; POINT TO PHD
MOVZWL #SSS_NORMAL,PHD\$L_R0(R3) ; SKIP PC OVER CHMK INSTRUCTION
SET NORMAL COMPLETION FOR WAIT
EVENT EXECUTABLE ACTION ROUTINE
EVENTE: ; EVENT EXECUTABLE STATE CHANGE
EVENTF: PUSHAB B^SCH\$CHSE ; ACTIVATE WITH NO WAIT TIME ACCOUNTING
BRB SCH\$UNWAIT ; MAKE UNWAIT EXIT THROUGH CHSE
; AND FALL INTO UNWAIT

M 2

004D 270 .SBTTL SCH\$UNWAIT - DECREMENT COUNT IN WAIT QUEUE
 004D 271 :++
 004D 272 : FUNCTIONAL DESCRIPTION:
 004D 273 : SCH\$UNWAIT DECREMENTS THE NUMBER OF PROCESSES IN THE WAIT
 004D 274 : QUEUE SELECTED BY THE SPECIFIED PCB AND STATE VALUE.
 004D 275 :
 004D 276 : CALLING SEQUENCE:
 004D 277 : BSB/JSB SCH\$UNWAIT
 004D 278 :
 004D 279 : INPUT PARAMETERS:
 004D 280 : R1 - STATE NUMBER (PRESERVED)
 004D 281 : R2 - UNUSED (PRESERVED)
 004D 282 : R4 - PCB ADDRESS (PRESERVED)
 004D 283 :
 004D 284 :
 004D 285 : IMPLICIT INPUTS:
 004D 286 : PCB LOCATED BY ADDRESS IN R4
 004D 287 :
 004D 288 : IMPLICIT OUTPUTS:
 004D 289 : COUNT IN WAIT QUEUE HEADER IS DECREMENTED IF STATE IS A WAIT
 004D 290 : STATE.
 004D 291 :
 004D 292 :--
 004D 293 :
 004D 294 SCH\$UNWAIT:: : DECREMENT PROPER WAIT COUNT
 1A 7B'AF 51 E1 004D 295 BBC R1,B^WAITMSK,20\$: SKIP OUT IF NOT WAIT STATE
 51 03 B1 0052 296 CMPW #SCH\$C_CEF,R1 : CHECK FOR COMMON EVENT FLAG WAIT
 16 13 0055 297 BEQL 30\$: CEF WAIT
 50 00000000'EF41 9E 005A 298 MULL #WQH\$C_LENGTH,R1 : COMPUTE BYTE INDEX TO WQ HDR
 08 A0 B7 0062 299 MOVAB L^SCH\$AQ_WQHDR[R1],R0 : COMPUTE ADDRESS OF WAIT Q HEADER
 0118 C4 0000'CF D0 0065 300 10\$: DECW WQH\$W_WQ\$CNT(R0) : DECREMENT WAIT QUEUE COUNT
 05 006C 301 MOVL W^EXE\$GL_ABSTIM,PCBSL_WAITIME(R4) ; Record event time
 006D 302 RSB ; RETURN
 50 2E A4 9A 006D 303 :
 50 50 A440 D0 0071 304 30\$: MOVZBL PCBSB_WEFC(R4),R0 : WAIT CLUSTER NUMBER
 50 14 C0 0076 305 MOVL PCBSL_EFC\$[R4][R0],R0 : GET CLUSTER ADDRESS
 E7 11 0079 306 ADDL #CEB\$[_WQFL,R0] : POINT TO WAIT QUEUE HEADER
 007B 307 BRB 10\$: GO DECREMENT WAIT COUNT
 007B 308 :
 007B 309 GMASK CEF : COMMON EVENT FLAG
 007B 310 GMASK LEF : LOCAL EVENT FLAG WAIT
 007B 311 GMASK LEFO : LOCAL EVENT FLAG WAIT
 007B 312 GMASK HIB : HIBERNAT WAIT
 007B 313 GMASK HIBO : HIBERNATE WAIT
 007B 314 GMASK FPG : FREE PAGE WAIT
 007B 315 GMASK COLPG : COLLISION PAGE WAIT
 007B 316 GMASK PFW : PAGE FAULT WAIT
 007B 317 GMASK SUSP : SUSPENDED WAIT
 007B 318 GMASK SUSPO : SUSPENDED WAIT
 007B 319 GMASK MWAIT : MUTEX WAIT
 00000FFE 007B 320 WAITMSK:.LONG WAITST : MASK OF WAIT STATES
 007F 321

 RSE
 VA
 The
 730
 26
 Mac
 --
 -S
 -S
 TO
 11
 The
 MAI

007F 323 .SBTTL SITUATIONAL PRIORITY INCREMENT TABLE
007F 324 :
007F 325 : FIXED DATA:
007F 326 : SITUATIONAL PRIORITY INCREMENT TABLE
007F 327 : (INDEXED BY PRIORITY INCREMENT CLASS)
007F 328 :
007F 329 :
007F 330 B_PINC:
00 007F 331 .BYTE 0 : CLASS 0 - NONE
02 0080 332 .BYTE 2 : CLASS 1 - I/O COMPLETE
03 0081 333 .BYTE 3 : CLASS 2 - RESOURCE AVAIL
04 0082 334 .BYTE 4 : CLASS 3 - TERM OUTPUT COMP
06 0083 335 .BYTE 6 : CLASS 4 - TERM INPUT COMP
0084 336 :
0084 337 :
00003000 0084 338 EXESTATE: : EXECUTABLE STATE MASK
0084 339 .LONG <1@SCH\$C_COM>!<1@SCH\$C_COM0>
0088 340 :

0088 342 .SBTTL SCH\$CHSE - CHANGE STATE TO EXECUTABLE
 0088 343 :++
 0088 344 : FUNCTIONAL DESCRIPTION:
 0088 345 : SCH\$CHSE CHANGES THE STATE OF A PROCESS, AS REPRESENTED BY
 0088 346 : ITS PCB, TO AN EXECUTABLE STATE. THE RESCHEDULING INTERRUPT
 0088 347 : WILL BE TRIGGERED IF THE PROCESS IS RESIDENT AND HAS A PRIORITY
 0088 348 : GREATER THAN THAT OF THE CURRENTLY EXECUTING PROCESS. A
 0088 349 : PRIORITY INCREMENT CLASS NUMBER SUPPLIED AS A REGISTER CONTAINED
 0088 350 : ARGUMENT IS USED TO COMPUTE THE NEW PROCESS PRIORITY FROM ITS
 0088 351 : BASE PRIORITY.
 0088 352 :
 0088 353 : CALLING SEQUENCE:
 0088 354 : BSB/JSB SCH\$CHSE
 0088 355 :
 0088 356 : INPUT PARAMETERS:
 0088 357 : R0 - NEW PRIORITY (SCH\$CHSEP ONLY)
 0088 358 : R2 - PRIORITY INCREMENT CLASS NUMBER (SCH\$CHSF ONLY)
 0088 359 : 0 => NO INCREMENT (PAGEFAULT I/O COMPLETION)
 0088 360 : 1 => NON-TERMINAL I/O COMPLETION
 0088 361 : 2 => RESOURCE AVAILABILITY
 0088 362 : 3 => TERMINAL OUTPUT COMPLETION
 0088 363 : 4 => TERMINAL INPUT COMPLETION
 0088 364 : R4 - PCB ADDRESS
 0088 365 :
 0088 366 : IMPLICIT INPUTS:
 0088 367 : SCH\$AQ_COMT - COMPUTE QUEUE HEADERS FOR COM,COMO STATES
 0088 368 : SCH\$GB_PRI - CURRENT PROCESS PRIORITY.
 0088 369 :
 0088 370 :
 0088 371 : OUTPUT PARAMETERS:
 0088 372 : R2 - R2, PRIORITY INCREMENT CLASS NUMBER IF SCH\$CHSE. (PRESERVED)
 0088 373 : R3 - R3 (PRESERVED)
 0088 374 :
 0088 375 : IMPLICIT OUTPUTS:
 0088 376 : SCH\$AQ_COMM - VECTOR OF COMPUTE QUEUE HEADERS.
 0088 377 : SCH\$GL_COMQS - COMPUTE QUEUE SUMMARY BIT VECTOR.
 0088 378 :
 0088 379 : COMPLETION CODES:
 0088 380 : NONE
 0088 381 :
 0088 382 : SIDE EFFECTS:
 0088 383 : THE PCB SPECIFIED IS REMOVED FROM ITS PRESENT STATE QUEUE
 0088 384 : AND INSERTED IN THE APPROPRIATE COMPUTE QUEUE, COM OR COMO.
 0088 385 : AT THE PRIORITY COMPUTED FOR THE SPECIFIED SITUATION CLASS.
 0088 386 : THE SUMMARY BIT FOR THE DESTINATION STATE QUEUE IS SET TO
 0088 387 : NOTE THAT IT IS OCCUPIED.
 0088 388 : IF THE NEW PRIORITY FOR THE PROCESS IS GREATER THAN THAT OF
 0088 389 : CURRENT PROCESS AND IT IS RESIDENT, THE RESCHEDULING INTERRUPT
 0088 390 : WILL BE TRIGGERED.
 0088 391 :
 0088 392 :--
 0088 393 : SCH\$CHSE:
 50 2F A4 50 D4 0088 394 CLRL R0 : CHANGE TO EXECUTABLE STATE
 0B A4 50 83 008A 395 SUBB3 B_PINC[R2],PCBSB_PRIB(R4),R0 : CLEAR HIGH SUM BITS FOR ADDB
 04 15 0091 396 CMPB R0,PCBSB_PRI(R4) ; ADD PRIORITY INCR
 50 0B A4 90 0097 397 BLEQ 10\$; CHECK FOR > CURRENT PRI
 0098 MOVB PCBSB_PRI(R4),R0 ; NO
 0099 : KEEP CURRENT PRIORITY INSTEAD

50	2F	A4	50	D4	0088	394	CLRL	R0	: CHANGE TO EXECUTABLE STATE
0B	A4	50	83	008A	395	SUBB3	B_PINC[R2],PCBSB_PRIB(R4),R0	: CLEAR HIGH SUM BITS FOR ADDB	
04	15	0091	396	CMPB	R0,PCBSB_PRI(R4)	; ADD PRIORITY INCR			
50	0B	A4	90	0097	397	BLEQ	10\$; CHECK FOR > CURRENT PRI	
					398	MOVB	PCBSB_PRI(R4),R0	; NO	
								; KEEP CURRENT PRIORITY INSTEAD	

10 50 91 009B 399 10\$: CMPB R0,#16 ; CHECK FOR RESULT >15
 04 18 009E 400 BGEQ SCH\$CHSEP ; YES, USE COMPUTED VALUE
 50 2F A4 90 00A0 401 MOVB PCB\$B_PRIB(R4),R0 ; KEEP AT BASE IF LESS
 00A4 402
 00A4 403 :
 00A4 404 : SCH\$CHSEP - SUB-ENTRY POINT WITH PRIORITY PRECOMPUTED IN R0
 00A4 405 :
 00A4 406 :
 00A4 407 SCH\$CHSEP:: ENTRY WITH PRIO IN R0
 51 12 DB 00A4 408 MFPR #PRS IPL,R1
 08 51 D1 00A7 409 CMPL R1,#IPLS_SYNCH
 62 19 00AA 410 BLSS BADIPL
 51 64 0F 00AC 411 REMQUE (R4),R1
 1C 12 00AF 412 BNEQ 10\$
 51 2C A4 3C 00B1 413 MOVZWL PCB\$W_STATE(R4),R1
 13 CB AF 51 E1 00B5 414 BBC R1,EXESTATE,10\$
 51 0B A4 9A 00BA 415 MOVZBL PCB\$B_PRI(R4),R1
 03 2C A4 E9 00BE 416 BLBC PCB\$W_STATE(R4),5\$
 51 20 C0 00C2 417 ADDL #32,RT
 00 00000000'EF 51 E5 00C5 418 5\$: BBCC R1,L^SCH\$GL_COMQS,10\$
 0B A4 50 90 00CD 419 10\$: MOVB R0,PCBSB_PRI(R4)
 51 0C D0 00D1 420 MOVL #SCH\$C_COM,R1
 12 24 A4 E8 00D4 421 BLBS PCB\$L_STS(R4),20\$
 51 D6 00D8 422 INCL R1
 50 20 C0 00DA 423 ADDL2 #32,R0
 00 00000000'EF 50 E2 00DD 424 BBSS R0,L^SCH\$GL_COMQS,15\$
 01CD 30 00E5 425 15\$: BSBW SCH\$SWP_WAKE
 14 11 00E8 426 BRB 35\$
 50 00000000'EF 91 00EA 427 20\$: CMPB L^SCH\$GB_PRI,R0
 03 19 00F1 428 BLSS 30\$
 00F3 429 SOFTINT #IPLS_SCHED
 00 00000000'EF 50 E2 00F6 430 30\$: BBSS R0,L^SCH\$GL_COMQS,35\$
 2C A4 51 B0 00FE 431 35\$: MOVW R1,PCBSW_STATE(R4)
 51 00000000'EF40 7E 0102 432 MOVAQ L^SCH\$AQ_COMT[R0],R1
 91 64 0E 010A 433 INSQUE (R4),@(RT)+
 010D 434 RSB
 010E 435
 010E 436 BADIPL: BUG_CHECK BADRSEIPL,FATAL ; BAD IPL AT ENTRANCE TO RSE
 0112 437

0112 439 .SBTTL SWPO - SWAP OUT SIMPLE NON-EXECUTABLE
0112 440 :
0112 441 : SWPO - SWAP OUT ACTION ROUTINE FOR SIMPLE NON-EXECUTABLE STATES
0112 442 :
0112 443 SWPO: : NON-EXECUTABLE OUTSWAP
FF38 30 0112 444 BSBW SCH\$UNWAIT : REMOVE FROM WAIT QUEUE
2C A4 B6 0115 445 INCW PCB\$W STATE(R4) : UPDATE STATE NUMBER
51 64 0F 0118 446 REMQUE (R4),R1 : REMOVE FROM WAIT QUEUE
10 B0 64 0E 011B 447 INSQUE (R4),@WQH\$L WQBL+WQH\$C LENGTH(R0) : INSERT AT TAIL OF QUEUE
14 A0 B6 011F 448 INCW WQH\$W_WQ\$CNT+WQH\$C_LENGTH(R0) : NOTE COUNT IN WAIT QUEUE
05 0122 449 RSB ; EXIT
0123 450 :
0123 451 :
0123 452 : SWPOE - SWAP OUT EXECUTABLE ACTION ROUTINE
0123 453 :
50 0B A4 9A 0123 454 SWPOE: MOVZBL PCB\$B_PRI(R4),R0 : GET PRIORITY
FF7A 31 0127 455 BRW SCH\$CH\$SEP : AND CHANGE TO COMO

012A 457 .SBTTL SCH\$QEND - QUANTUM END ROUTINE

012A 458

012A 459 :++

012A 460

012A 461 : FUNCTIONAL DESCRIPTION:
012A 462 SCH\$QEND IS CALLED BY THE TIMER WHEN THE QUANTUM FOR THE CURRENT
012A 463 PROCESS HAS BEEN EXHAUSTED. A NEW QUANTUM IS INITIALIZED
012A 464 THE PROCESS PLACED AT ITS BASE PRIORITY AND THE RESCHEDULING
012A 465 INTERRUPT TRIGGERED. A CHECK IS MADE FOR CPU TIME LIMIT EXPIRATION
012A 466 AND APPROPRIATE EXIT ASTS GENERATED WHEN THE LIMIT IS REACHED.
012A 467 THE AUTOMATIC WORKING SET SIZE LOGIC IS INVOKED IF ENABLED TO
012A 468 TRADEOFF WORKING SET SIZE AGAINST PAGEFAULT RATE.

012A 469

012A 470 : CALLING SEQUENCE:
012A 471 BSB/JSB SCH\$QEND

012A 472

012A 473 : INPUT PARAMETERS:
012A 474 R4 - PCB ADDRESS OF CURRENT PROCESS
012A 475 R5 - PROCESS HEADER ADDRESS

012A 476

012A 477 : IMPLICIT INPUTS:
012A 478 PCB OF CURRENT PROCESS
012A 479 PROCESS HEADER OF CURRENT PROCESS

012A 480

012A 481 : IMPLICIT OUTPUTS:
012A 482 PHDSW_QUANT - INITIALIZED TO A NEW QUANTUM
012A 483 PCBSV_INQUAN - INITIAL QUANTUM FLAG CLEARED

012A 484

012A 485 :--

012A 486

00 24 A4 03 E5 012A 487 SCH\$QEND:: : QUANTUM END ROUTINE

3C A5 00000000'EF B0 012F 488 BBCC #PCBSV_INQUAN,PCBSL_STS(R4),10\$: CLEAR INITIAL QUAN FLAG

0118 C4 0000'CF D0 0137 489 10\$: MOVW SCH\$GW_QUAN,PHDSW_Q0ANT(R5) : SET NEW QUANTUM

10 0B A4 91 013E 490 MOVL W^EXE\$GL_AB\$TIM,PCBSL_WAITIME(R4) : Record event time

26 19 0142 491 CMPB PCBSB_PRI(R4),#16 : CHECK FOR REAL-TIME

0144 492 BLSS 50\$: YES

0144 493

0144 494

0144 495 : CHECK FOR CPU TIME LIMIT EXPIRATION

0144 496

5C A5 D5 0144 497 TSTL PHDSL_CPULIM(R5) : IS THERE ANY LIMIT?

22 12 0147 498 BNEQ 60\$: YES, GO CHECK IT OUT

09 24 A4 18 E0 0149 499 40\$: BBS #PCBSV_DISAWS,PCBSL_STS(R4),45\$: BRANCH IF ADJUSTMENT DISABLED

53 0000'CF D0 014E 500 MOVL W\$CH\$GL_WSINC,R3 : ASSUME INCREMENT

02 13 0153 501 BEQL 45\$: BR IF NO AUTO WS ADJUSTMENT

49 10 0155 502 BSBB WSADJUST : ELSE GO DO IT

00000000'EF D5 0157 503 45\$: TSTL L\$CH\$GL_COMOQS : IS THERE ANY INSWAP PENDING?

08 13 015D 504 BEQL 47\$: NO

0B A4 2F A4 90 015F 505 MOVB PCBSB_PRIB(R4),PCBSB_PRI(R4) : YES, FORCE TO BASE PRIORITY

014E 30 0164 506 BSBW SCH\$SWP_WAKE : AND WAKE SWAPPER

0167 507 47\$: SOFTINT #IPLS_SCHED : TRIGGER RESCHEDULING INT

05 016A 508 50\$: RSB : AND RETURN

016B 509

016B 510

016B 511 : A non-zero limit exists, check for processor time expiration

016B 512 : If CPU time limit is exceeded then an additional amount of time will

016B 513 :

016B 514 : be allowed for each access mode. An AST will be issued to cause an
 016B 515 : exit for each of the access modes. The additional time allowance will
 016B 516 : be provided for each access mode.
 016B 517 :
 016B 518 :
 50 38 A5 5C A5 C3 016B 519 60\$: SUBL3 PHDSL_CPUTIM(R5),PHDSL_CPUTIM(R5),R0 : HAS LIMIT BEEN REACHED
 D6 1F 0171 520 BLSSU 40\$; NO, CONTINUE NORMALLY
 0173 521 :
 0173 522 : CPU LIMIT HAS EXPIRED, AN AST WILL BE SENT TO NOTIFY THE PROCESS
 0173 523 :
 50 0000'CF CO 0173 524 ADDL2 W^SGN\$GL_EXTRACPU,R0 : COMPUTE TOTAL AMOUNT OF EXTRA TIME
 5C A5 50 CO 0178 525 ADDL2 R0,PHDSL_CPUTIM(R5) : GIVE EXTRA TIME FOR CLEANUP
 0110 C5 50 CO 017C 526 ADDL2 R0,PHDSL_EXTRACPU(R5) : AND RECORD AMOUNT OF EXTRA TIME
 50 60 A5 9E 0181 527 MOVAB PHDSB_CPUTIM(R5),R0 : GET ADDRESS OF AST ACCESS MODE
 C1 AF 9F 0185 528 PUSHAB 40\$: SET RETURN ADDRESS
 53 20AC BF 3C 0188 529 MOVZWL #SSS_EXCPUTIM,R3 : PASS EXIT STATUS TO SENDAST
 00BD 30 018D 530 SCH\$FORCEDEXIT::: BSBW SENDAST : SEND AST TO PROCESS
 0190 531 :
 0190 532 : CPU TIME EXPIRATION AST HANDLER
 0190 533 :
 00 0000 0190 534 CPUABRT: WORD 0 : NULL ENTRY MASK
 00 BC 0192 535 CHMK S^#ASTEXIT : EXIT FROM AST ROUTINE (CLEAR AST)
 0194 536 10\$: SEXIT_S 4(AP) : EXIT TO INVOKE EXIT HANDLERS
 F4 11 019E 537 BRB 10\$: JUST IN CASE
 01A0 538 :
 01A0 539 :
 01A0 540 : Adjust working set size automatically to achieve desired tradeoff
 01A0 541 : between page fault rate and working set size. There are two page
 01A0 542 : fault rate thresholds: SCH\$GL_PFRATL, the lower threshold and
 01A0 543 : SCH\$GL_PFRATH, the higher threshold. Each time SCH\$QEND is invoked,
 01A0 544 : the page fault rate is computed and compared with these thresholds.
 01A0 545 : If it is above the high threshold the working set size is increased
 01A0 546 : by SCH\$GW_WSINC and if the rate is below the lower threshold, the
 01A0 547 : working set size is decreased by SCH\$GW_WSDEC. The actual adjustment
 01A0 548 : is performed by a normal kernel mode AST.
 01A0 549 :
 01A0 550 :
 01A0 551 : Automatic adjustment of working set size is constrained by the values:
 01A0 552 : SCH\$GW_AWSMIN and WSEXTENT per process that establish upper and lower
 01A0 553 : values for automatic working set size adjustment. Working set size
 01A0 554 : adjustment is further constrained by the process quota.
 01A0 555 :
 01A0 556 :
 01A0 557 : R3 - Working set increment
 01A0 558 :
 50 0100 C5 C3 01A0 559 WSADJUST: SUBL3 PHDSL_TIMREF(R5),- : AUTO-ADJUST WORKING SET SIZE
 38 A5 01A0 560 PHDSL_CPUTIM(R5),R0 : COMPUTE DELTA-T
 02 12 01A4 561 BNEQ 10\$: BR IF NON-ZERO
 50 D6 01A7 562 INCL R0 : ELSE FORCE TO ONE FOR DIVIDE
 0000'CF 50 D1 01AB 563 10\$: CMPL R0,W^SCH\$GL_AWSTIME : IS THIS A MEANINGFUL INTERVAL?
 3F 19 01B0 564 BLSS NOADJUST : NO, TRY AGAIN LATER
 00FC C5 C3 01B2 565 SUBL3 PHDSL_PFLREF(R5),- : COMPUTE DELTA-PGFLT
 51 4C A5 01B6 566 PHDSL_PAGEFLTS(R5),R1 :
 00FC C5 4C A5 D0 01B9 567 MOVL PHDSL_PAGEFLTS(R5),PHDSL_PFLREF(R5) : SAVE NEW PAGE FAULT REF
 0100 C5 38 A5 D0 01BF 568 MOVL PHDSL_CPUTIM(R5),PHDSL_TIMREF(R5) : AND SAVE CPUTIME REF
 570

51 000003E8 8F C4 01C5 571 MULL #1000,R1 ; MULTIPLY BY SCALE FACTOR
 51 50 C6 01CC 572 DIVL R0,R1 AND COMPUTE PAGEFLTS/10SEC
 00F8 C5 51 D0 01CF 573 MOVL R1,PHDSL_PFLTRATE(R5) SAVE CURRENT RATE
 0000'CF 51 D1 01D4 574 CMPL R1,W\$CH\$GL_PFRATH ARE WE ABOVE HIGH THRESHOLD?
 17 18 01D9 575 BGEQ ADJUSTUP YES,
 53 0000'CF CE 01DB 576 MNEGL W\$CH\$GL_WSDEC,R3 NO, GET DECREMENT VALUE
 0000'CF 51 D1 01E0 577 CMPL R1,W\$CH\$GL_PFRATL ARE WE BELOW LOW THRESHOLD?
 0A 18 01E5 578 BGEQ NOADJUST NO, IN DEAD BAND -- NOTHING TO DO
 0000'CF 36 A4 B1 01E7 579 CMPW PCB\$W_PPGCNT(R4),W\$CH\$GW_AWSMIN ; ARE WE AT LOWER WS LIMIT?
 02 1B 01ED 580 BLEQU NOADJUST YES, NOTHING TO DO
 39 11 01EF 581 BRB ADJUST
 01F1 582 NOADJUST:
 05 01F1 583 RSB
 01F2 584 ADJUSTUP:
 51 18 A5 08 A5 A3 01F2 585 SUBW3 PHDSW_WSLIST(R5),PHDSW_W\$QUOTA(R5),R1
 00000000'EF 50 50 A5 3C 01F8 586 01F8 587 MOVZWL PHDSW_WSSIZE(R5),R0 ; ASSUME HIGH LIMIT WILL BE QUOTA
 0000'CF D1 01FC 588 CMPL W\$CH\$GL_BORROWLIM,L\$CH\$GL_FREECNT ; GET CURRENT WORKING SET SIZE
 06 1A 0205 589 BGTRU 10S ; ARE THERE LOTS OF FREE PAGES?
 51 16 A5 08 A5 A3 0207 590 SUBW3 PHDSW_WSLIST(R5),PHDSW_W\$EXTENT(R5),R1
 020D 591 10\$: CMPW R0,R1 ; BRANCH IF MEMORY IS AT A PREMIUM
 51 50 B1 020D 592 10\$: BGTRU NOADJUST ; ALLOW LARGER GROWTH SIZE
 DF 1A 0210 593 ADDW3 PCB\$W_GPGCNT(R4),PCBSW_PPGCNT(R4),R1 ; ARE WE AT MAXIMUM SIZE?
 51 36 A4 34 A4 A1 0212 594 CMPW R0,R1 ; YES, CAN'T GO ANY LARGER
 51 50 B1 0218 595 2C 1F 021B 596 BLSSU WSERR ; GET CURRENT PHYSICAL SIZE
 52 50 FE 8F 78 021D 597 ASHL #2,R0,R2 ; Be sure that pages in use don't exceed WS
 50 52 A2 0222 598 SUBW2 R2,R0 ; BRANCH IF WS SMALLER THAN PAGES IN USE
 50 51 B1 0225 599 CMPW R1,R0 ; Compute 75% of WSSIZE as page threshold
 C7 1F 0228 600 BLSSU NOADJUST ; If threshold not exceeded,
 50 61 A5 9E 022A 601 ADJUST: MOVAB PHDSB_AWSMODE(R5),R0 ; skip WS adjustment
 1D 10 022E 602 BSBB SENDAST ; GET ADDRESS OF AST ACCESS MODE
 0230 603 604 ADJWS: WORD 0 ; SEND AST TO PROCESS
 51 00000000'9F D0 0230 604 ADJWS: .WORD 0 ; GET PHD ADDRESS SO
 61 A1 94 0232 605 MOVL @#CTL\$GL_PHD,R1 ; ACCESS MODE FLAG CAN BE RESET
 0239 606 CLRBL PHDSB_AWSMODE(R1) ; ADJUST BY PARAMETER IN AST ARGLIST
 023C 607 SADJWSL_S 4(AP) ; AND RETURN
 04 0248 608 RET
 0249 609
 0249 610 WSERR: BUG_CHECK WSSIZEERR,FATAL ; WORKING SET SIZE CALC IN ERROR

024D 612 .SBTTL SENDAST - Send AST to process
 024D 613 ++
 024D 614 : FUNCTIONAL DESCRIPTION: SENDAST IS CALLED BY SCH\$QEND TO SEND ASTS TO THE
 024D 615 : PROCESS THAT INVOKE FUNCTIONS UNAVAILABLE TO THE ENVIRONMENT OF SCH\$QEND.
 024D 616 : THESE INCLUDE ADJUSTING THE WORKING SET AND EXITTING.
 024D 617
 024D 618 INPUT PARAMETERS:
 024D 619
 024D 620 R0 - ADDRESS OF ACCESS MODE FOR AST
 024D 621 : (NEGATIVE CONTENTS PREVENT SENDING AST)
 024D 622 R3 - AST PARAMETER
 024D 623 R4 - PCB ADDRESS
 024D 624 (SP) - AST ADDRESS
 024D 625 4(SP) - RETURN ADDRESS FOR THIS SUBROUTINE
 024D 626 --
 024D 627 SENDAST:
 50 DD 024D 628 PUSHL R0 : SAVE ADDRESS OF ACCESS MODE
 53 DD 024F 629 PUSHL R3 : AND AST PARAMETER
 60 95 0251 630 TSTB (R0) : CHECK VALUE OF ACCESS MODE
 37 19 0253 631 BLSS 10\$: DO NOT QUEUE AST IF NEGATIVE
 32 24 A4 01 E0 0255 632 BBS #PCBSV_DELPEN,PCBSL_STS(R4),10\$: NOR IF MARKED FOR DELETE
 51 1C 3C 025A 633 MOVZWL #ACB\$C_LENGTH,R1 : SET SIZE REQUIRED
 FDAO' 30 025D 634 BSBW EXESAL\$NONPAGED : ALLOCATE A BLOCK
 29 50 E9 0260 635 BLBC R0,10\$: NONE, TRY LATER
 0A A2 02 90 0263 636 MOVB #DYN\$C_ACB,ACBSB_TYPE(R2) : SET TYPE OF STRUCTURE
 08 A2 51 B0 0267 637 MOVW R1,ACBSW_SIZE(R2) : AND SIZE OF STRUCTURE
 14 A2 8E D0 026B 638 MOVL (SP)+,ACBSL_ASTPRM(R2) : AND AST PARAMETER VALUE
 0B A2 00 BE 90 026F 639 MOVB @SP,ACBSB_RMOD(R2) : SET ACCESS MODE FOR AST
 10 A2 8E D0 0274 640 DECB @SP+ : INDICATE SUCCESS FOR THIS ACCESS MODE
 0C A2 60 A4 D0 027A 641 MOVL (SP)+,ACBSL_AST(R2) : SET AST ADDRESS
 30 BB 027F 642 MOVL PCBSL_PID(R4),ACBSL_PID(R2) : SET PID FOR AST
 55 52 D0 0281 643 PUSHR #^M<R4,R5> : SAVE REGS FOR QAST
 52 D4 0284 644 MOVL R2,R5 : SET ADDRESS OF ACB
 FD77' 30 0286 645 CLRL R2 : NULL PRIORITY INCREMENT
 30 BA 0289 647 BSBW SCH\$QAST : QUEUE AST FOR PROCESS
 05 028B 648 POPR #^M<R4,R5> : RESTORE PCB,PHD ADDRESSES
 028C 649 RSB : EXIT
 028C 650 : Error path if nonpaged pool allocation fails or if AST access mode is
 028C 651 : negative, indicating either an AST in progress (for automatic working
 028C 652 : set adjustment) or all access modes are done (for CPU time limit expiration)
 028C 653
 5E 0C C0 028C 654 10\$: ADDL #12,SP : CLEAN PARAMETERS FROM STACK
 05 028F 655 RSB : AND EXIT
 0290 656

0290 658
 0290 659 .SBTTL SCH\$WAKE - WAKE PROCESS INTERNAL
 0290 660 ++
 0290 661 : FUNCTIONAL DESCRIPTION:
 0290 662 : SCH\$WAKE WAKES THE PROCESS SPECIFIED BY THE PID SUPPLIED.
 0290 663 :
 0290 664 : CALLING SEQUENCE:
 0290 665 : BSB/JSB SCH\$WAKE
 0290 666 :
 0290 667 : INPUT PARAMETERS:
 0290 668 : R1 - PID OF PROCESS TO WAKE
 0290 669 :
 0290 670 : OUTPUT PARAMETERS:
 0290 671 : R0 - COMPLETION STATUS CODE
 0290 672 : R4 - PCB ADDRESS OF PROCESS AWAKENED
 0290 673 :
 0290 674 : COMPLETION CODES:
 0290 675 : SSS_NORMAL - NORMAL SUCCESSFUL COMPLETION STATUS
 0290 676 : SSS_NONEPR - NONEXISTENT PROCESS (INVALID PID)
 0290 677 :
 0290 678 : ENVIRONMENT:
 0290 679 : IPL = IPL\$_SYNCH
 0290 680 :
 0290 681 :--
 0290 682 SCH\$WAKE:: : WAKE PROCESS INTERNAL
 54 54 51 3C 0290 683 MOVZWL R1,R4 : GET PROCESS INDEX (PIX)
 0000'DF44 D0 0293 684 MOVL @W\$SCH\$GL_PCBVEC[R4],R4 : LOOK UP PCB ADDRESS
 60 A4 51 D1 0299 685 CMPL R1,PCBSL_PID(R4) : VERIFY PID
 10 12 029D 686 BNEQ 30\$: REPORT ERROR
 00 24 A4 0C E2 029F 687 BBSS #PCBS\$V_WAKEOPEN,PCBSL_STS(R4),10\$; SET WAKE PENDING
 02A4 688 10\$:
 52 02 9A 02A4 689 MOVZBL #PRI\$_RESAVL,R2 : SET PRIORITY INCREMENT CLASS
 02A7 690 RPTEVT WAKE : REPORT WAKE EVENT
 50 01 3C 02AB 691 MOVZWL #SSS_NORMAL,R0 : SET SUCCESS CODE
 05 02AE 692 20\$: RSB : RETURN
 02AF 693
 50 08E8 8F 3C 02AF 694 30\$: MOVZWL #SSS_NONEPR,R0 : SET NONEXISTENT PROCESS STATUS
 05 02B4 695 RSB :

02B5 697 .SBTTL SCH\$SWPWAKE - WAKE SWAPPER PROCESS
 02B5 698 :++
 02B5 699 : FUNCTIONAL DESCRIPTION:
 02B5 700 : SCH\$SWPWAKE AWAKENS THE SWAPPER PROCESS TO PERFORM SOME OPERATION.
 02B5 701 :
 02B5 702 : CALLING SEQUENCE:
 02B5 703 : BSB/JSB SCH\$SWPWAKE
 02B5 704 :
 02B5 705 : INPUT PARAMETERS:
 02B5 706 : NONE
 02B5 707 :
 02B5 708 : OUTPUT PARAMETERS:
 02B5 709 : R0-R4 PRESERVED
 02B5 710 :
 02B5 711 : SIDE EFFECTS:
 02B5 712 : A WAKE EVENT IS GENERATED FOR THE SWAPPER PROCESS WHICH CAN
 02B5 713 : CAUSE THE PROCESSOR TO BE RESCHEDULED.
 02B5 714 :
 02B5 715 :--
 02B5 716 :
 02B5 717 SCH\$SWPWAKE:: : WAKE SWAPPER PROCESS
 00000000'EF D5 02B5 718 TSTL L^SCH\$GL_COMOQS : ANY INSWAP CANDIDATES?
 28 12 02BB 719 BNEQ 10\$: YES, MUST WAKE SWAPPER THEN
 00000000'EF D1 02BD 720 CMPL L^SCH\$GL_MFYCNT,L^SCH\$GL_MFYLIM : ARE THERE MODIFIED PAGES TO WRITE?
 1B 18 02C8 721 BGEQ 10\$: YES, MUST WAKE SWAPPER THEN
 00000000'EF D1 02CA 722 CMPL L^SCH\$GL_FREECNT,L^SCH\$GL_FREELIM : DO WE NEED FREE PAGES?
 0E 19 02D5 723 BLSS 10\$: YES, MUST WAKE SWAPPER THEN
 00000000'EF B5 02D7 724 TSTW L^SCH\$GW_DELPHDCT : ARE THERE DELETED HEADERS TO PURGE?
 06 12 02DD 725 BNEQ 10\$: YES, MUST WAKE SWAPPER THEN
 0000'CF D5 02DF 726 TSTL W^EXE\$GL_PFATIM : WAS THERE A POWER FAIL RECOVERY?
 13 13 02E3 727 BEQL 20\$: BR IF NONE
 00000000'EF 95 02E5 728 10\$: TSTB L^SCH\$GB_SIP : SWAPPER ALREADY BUSY?
 0B 12 02EB 729 BNEQ 20\$: BR IF YES
 1F BB 02ED 730 PUSHR #^M<R0,R1,R2,R3,R4> : SAVE R0-R4
 51 0000'CF D0 02EF 731 MOVL W^SCH\$GL_SWPPID,R1 : GET PID OF SWAPPER
 9A 10 02F4 732 BSBB SCH\$WAKE : AND AWAKEN IT
 1F BA 02F6 733 POPR #^M<R0,R1,R2,R3,R4> : RESTORE R0-R4
 05 02F8 734 20\$: RSB : AND RETURN TO CALLER
 02F9 735 :
 02F9 736 .END

ACB\$B_RMOD	= 0000000B	PCB\$W_STATE	= 0000002C
ACB\$B_TYPE	= 0000000A	PHD\$B_AWSMODE	= 00000061
ACB\$C_LENGTH	= 0000001C	PHD\$B_CPMODE	= 00000060
ACB\$L_AST	= 00000010	PHD\$L_CPLIM	= 0000005C
ACB\$L_ASTPRM	= 00000014	PHD\$L_CPUTIM	= 00000038
ACB\$L_PID	= 0000000C	PHD\$L_EXTRACPU	= 00000110
ACB\$W_SIZE	= 00000008	PHD\$L_PAGEFLTS	= 0000004C
ACTION	0000001A R 03	PHD\$L_PC	= 000000C0
ADJUST	0000022A R 03	PHD\$L_PFLREF	= 000000FC
ADJUSTUP	000001F2 R 03	PHD\$L_PFLTRATE	= 000000F8
ADJWS	00000230 R 03	PHD\$L_RO	= 00000088
ASTEXIT	= 00000000	PHD\$L_TIMREF	= 00000100
BADIPL	0000010E R 03	PHD\$W_QUANT	= 0000003C
BUGS_BADRSEIPL	***** X 03	PHD\$W_WSEXTENT	= 00000016
BUGS_ILLEVNUM	***** X 03	PHD\$W_WSLIST	= 00000008
BUGS_WSSIZEERR	***** X 03	PHD\$W_WSQUOTA	= 00000018
B_PINC	0000007F R 03	PHD\$W_WSSIZE	= 00000050
CEB\$L_WQFL	= 00000014	PRS_IPL	= 00000012
CEFEVT	00000038 R 03	PRS_SIRR	= 00000014
CPUABRT	00000190 R 03	PRI\$_RESAVL	= 00000002
CTL\$GL_PHD	***** X 03	RSE	= 00000034 R 03
DYN\$C_ACB	= 00000002	SCH\$AQ_COMT	***** X 03
EVENTE	0000004A R 03	SCH\$AQ_WQHDR	***** X 03
EVENTF	0000004A R 03	SCH\$CHSE	00000088 RG 03
EVT\$_AST	= 00000000 G	SCH\$CHSEP	000000A4 RG 03
EVT\$_COLPGA	= 00000000 G	SCH\$C_CEF	= 00000003
EVT\$_EVENT	= 00000001 G	SCH\$C_COLPG	= 00000001
EVT\$_FPGA	= 00000004 G	SCH\$C_COM	= 0000000C
EVT\$_PF\$COM	= 00000007 G	SCH\$C_COMO	= 0000000D
EVT\$_RESUME	= 00000006 G	SCH\$C_FPG	= 0000000B
EVT\$_SETPRI	= 00000008 G	SCH\$C_HIB	= 00000007
EVT\$_SWPOUT	= 00000009 G	SCH\$C_HIBO	= 00000008
EVT\$_WAKE	= 00000005 G	SCH\$C_LEF	= 00000005
EVTCTR	= 0000000B	SCH\$C_LEFO	= 00000006
EXE\$ALONONPAGED	***** X 03	SCH\$C_MWAIT	= 00000002
EXE\$GL_ABSTIM	***** X 03	SCH\$C_PFW	= 00000004
EXE\$GL_PFA\$TIM	***** X 03	SCH\$C_SUSP	= 00000009
EXESTATE	00000084 R 03	SCH\$C_SUSPO	= 0000000A
IPL\$_SCHED	= 00000003	SCH\$F\$ORCEDEXIT	0000018D RG 03
IPL\$_SYNCH	= 00000008	SCH\$GB_PRI	***** X 03
LEFEVT	0000003C R 03	SCH\$GB_SIP	***** X 03
MAXEVT	= 0000000A	SCH\$GL_AWSTIME	***** X 03
NOADJUST	000001F1 R 03	SCH\$GL_BORROWLIM	***** X 03
PCB\$B_PRI	= 0000000B	SCH\$GL_COMOQS	***** X 03
PCB\$B_PRIB	= 0000002F	SCH\$GL_COMQS	***** X 03
PCB\$B_WEFC	= 0000002E	SCH\$GL_FREECNT	***** X 03
PCB\$L_EFCS	= 00000050	SCH\$GL_FREELIM	***** X 03
PCB\$L_PHD	= 0000006C	SCH\$GL_MFYCNT	***** X 03
PCB\$L_PID	= 00000060	SCH\$GL_MFYLIM	***** X 03
PCB\$L_STS	= 00000024	SCH\$GL_PCBVEC	***** X 03
PCB\$L_WAITIME	= 00000118	SCH\$GL_PFRATH	***** X 03
PCBSV\$_DELPEN	= 00000001	SCH\$GL_PFRATL	***** X 03
PCBSV\$_DISAWS	= 00000018	SCH\$GL_SWPPID	***** X 03
PCBSV\$_INQUAN	= 00000003	SCH\$GL_WSDEC	***** X 03
PCBSV\$_WAKEPEN	= 0000000C	SCH\$GL_WSINC	***** X 03
PCBSW\$_GPGCNT	= 00000034	SCH\$GW_AWSMIN	***** X 03
PCBSW\$_PPGCNT	= 00000036	SCH\$GW_DELPHDCT	***** X 03

RSE Symbol table

- REPORT SYSTEM EVENT

L 3

16-SEP-1984 01:06:34 VAX/VMS Macro V04-00
5-SEP-1984 03:47:04 [SYS.SRC]RSE.MAR;1

Page 20
(1)

SCH\$GW_QUAN	*****	X	03
SCH\$QAST	*****	X	03
SCH\$QEND	0000012A	RG	03
SCH\$RSE	00000000	RG	03
SCH\$SWP_WAKE	000002B5	RG	03
SCH\$UNWAIT	0000004D	RG	03
SCH\$WAKE	00000290	RG	03
SENDAST	0000024D	R	03
SGN\$GL_EXTRACPU	*****	X	03
SSS_EXCPUTIM	= 000020AC		
SSS_NONEXPR	= 00000BEB		
SSS_NORMAL	= 00000001		
ST	= 00000002		
STACT	0000001E	R	03
STET	= 00000000	R	02
STMSK	= 00001000		
SWPO	00000112	R	03
SWPOE	00000123	R	03
SYSSADJWSL	*****	GX	03
SYSSEXIT	*****	GX	03
WAITMSK	0000007B	R	03
WAITST	= 00000FFE		
WQHSC_LENGTH	= 0000000C		
WQHSL_WQBL	= 00000004		
WQHSW_WQCNT	= 00000008		
WSADJUST	000001A0	R	03
WSERR	00000249	R	03

+-----+
! Psect synopsis !
+-----+

PSECT name

```

. ABS .
$ABSS
AES2
AES1
      00000000 ( 0.) 00 ( 0.) NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
      00000000 ( 0.) 01 ( 1.) NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
      0000002C ( 44.) 02 ( 2.) NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
      000002F9 ( 761.) 03 ( 3.) NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

```

Performance indicators

Phase

Phase	Page	Page Time	Elapsed Time
Initialization	36	00:00:00.09	00:00:01.96
Command processing	123	00:00:00.48	00:00:05.87
Pass 1	331	00:00:10.62	00:00:34.14
Symbol table sort	0	00:00:01.68	00:00:04.35
Pass 2	140	00:00:02.57	00:00:09.68
Symbol table output	18	00:00:00.13	00:00:00.13
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	652	00:00:15.60	00:00:56.16

The working set limit was 1650 pages.
62765 bytes (123 pages) of virtual memory were used to buffer the intermediate code.

There were 60 pages of symbol table space allocated to hold 1060 non-local and 26 local symbols.
736 source lines were read in Pass 1, producing 19 object records in Pass 2.
26 pages of virtual memory were used to define 25 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

12
8
20

1119 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LI\$S:RSE/0BJ=0BJ\$S:RSE MSRC\$S:RSE/UPDATE=(ENH\$S:RSE)+EXECML\$S/LIB

0380 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

